

Communication Apparatus and Communication Method

BACKGROUND OF THE INVENTION

Field of the Invention

5 The present invention relates to a communication apparatus provided in an electronic device capable of transmitting and receiving information data via a digital interface, and a communication method of the apparatus.

10 Related Background Art

 Recently, apparatuses and systems for processing not only text information such as documents but also various information such as images and sounds are beginning to be extensively used with the improvement
15 of the processing capability of a central processing unit (CPU) using a computer or the like, the progress of a graphical operating system (OS) for operating hardware, the increase in capacity and the progress of digitization of communication information in a network,
20 and the development of information compression techniques.

 With the development of such multimedia technologies, it is becoming possible to transmit all types of data in all forms by all communication
25 protocols via a single digital interface (digital I/F). It is also becoming possible for an apparatus corresponding to one communication protocol and

09022979-021298

As an example of the digital I/F bus systems described above, a communication system has been proposed in which AV devices, such as a digital video tape recorder (to be referred to as a VTR hereinafter), a digital television receiver, and a tuner, and a personal computer (to be referred to as a PC hereinafter) are mutually connected by an IEEE1394 serial bus (to be referred to as 1394 hereinafter), and digital video signals and digital audio signals are transmitted and received between these electronic devices. An outline of this 1394 system will be described below.

Each digital device described above has a function of relaying digital data and control data on the 1394 serial bus. Also, a cable for the 1394 serial bus

includes three shielded twisted pair lines. Each twisted pair line is used to transfer protocol signals and data and supply electric power. Therefore, the whole system can operate even when a certain device is
5 turned off in the system.

The basic configuration of each digital device has an operation unit as a user interface, a display unit, a CPU for controlling the operation of the whole device, forming packets for communication, and holding
10 addresses, a digital I/F for the 1394 serial bus, and a switch unit for performing switching between a deck unit, a tuner unit, or a camera unit (neither is shown) and the digital I/F.

In this 1394 system, as shown in Fig. 2,
15 communication is performed at a predetermined communication cycle (125 μ s). Data having a time axis such as video data or audio data is transmitted by isochronous (synchronous) communication by which a transfer band is guaranteed at a fixed data rate.
20 Control data such as a control command is transmitted irregularly, where necessary, by asynchronous communication.

In communication like this, a cycle start packet exists at the beginning of each communication cycle,
25 and a period for transmitting a packet for isochronous communication is set subsequently to the cycle start packet. A plurality of channels of isochronous

09022979-021298

communication can be simultaneously performed by assigning channel numbers to packets for isochronous communication.

For example, when channel 1 is assigned to communication from the DVCR to the VTR, the DVCR transmits an isochronous communication packet of channel number 1 onto the bus immediately after the cycle start packet. Meanwhile, the VTR monitors packets on the bus and receives the packet assigned with channel number 1. In this manner isochronous communication is executed between the DVCR and the VTR.

Analogously, when channel number 2 is assigned to a packet from the DCAM to the PC, isochronous communication is executed between the DCAM and the PC by transmitting the packet of channel number 2 onto the bus after the packet of channel number 1, and the isochronous communications between channel 1 and channel 2 are performed parallel. A period from the completion of transmission of all isochronous communication packets in each communication cycle to the next cycle start packet is used in asynchronous communication.

Bus management by which the 1394 serial bus system described above can operate will be described below.

An apparatus serving as a bus manager previously checks the network structure and the connection states of all nodes and controls bus communication by defining

00022979-021200

each node ID and controlling isochronous communication.

That is, in the communication system as described above, when the power supply is turned on or when a new digital device is connected or a certain device is disconnected, node IDs (physical addresses #0, #1, #2, and #3 in Fig. 3) are automatically assigned to the individual devices (nodes) in accordance with their connection states by the following procedure based on an address program and an address table stored in an internal memory of the CPU, thereby automatically setting topology.

This node ID assignment procedure will be briefly described below. This procedure includes determination of the hierarchical structure of the system and assignment of physical addresses to the nodes.

Assume that the above digital devices, i.e., the PC, DVCR, VTR, and DCAM are nodes A, B, C, and D, respectively.

First, each node transmits to a partner node, to which this node is connected by the 1394 serial bus, information indicating that the partner is its parent. While giving priority to a node first transmitting this information to its partner, the parent-child relationship between the nodes in this system, i.e., the hierarchical structure of the system and a route node which is not a child of any other node are finally determined.

09022979-021298

More specifically, the node D informs the node C that the partner is a parent, and the node B informs the node A that the partner is a parent. If the node A informs the node C that the partner is a parent and the node C informs the node A that the partner is a parent, a node which first transmits the information to its partner is given priority. That is, if the transmission from the node C is earlier, the node A is regarded as a parent of the node C. As a consequence, the node A is not a child of any other node. If this is the case, the node A is a route node.

After the parent-child relationship between the digital devices is thus determined, assignment of physical addresses is performed. This physical address assignment is basically done in such a manner that parent nodes permit child nodes to perform address assignment and these child nodes permit themselves to perform address assignment from one connected to the smaller port number.

When the parent-child relationship is determined as above in the example shown in Fig. 3, the node A first permits the node B to perform address assignment. As a consequence, the node B assigns physical address #0 to itself. The node B sends this information onto the bus to inform the other nodes that "physical address #0 is already assigned".

Next, the node A permits the node C to perform

05022979-021298

address assignment, and the node C similarly permits the node D, i.e., the child of the node C, to perform address assignment. Consequently, the node D assigns physical address #1, next to physical address #0, to
5 itself, and sends this information onto the bus.

Thereafter, the node C assigns physical address #2 to itself and sends this information onto the bus. Finally, the node A assigns physical address #3 to itself and sends this information onto the bus.

10 A data transfer procedure will be described next.

Data transfer is enabled by assigning physical addresses as described above. In the 1394 serial bus system, however, arbitration of the bus use rights is performed by the route node prior to data transfer.
15 That is, in the 1394 as shown in Fig. 4, only data of one channel is transferred at a certain timing. Therefore, the bus use rights must be arbitrated first.

When each node wants to perform data transfer, the node requests its parent node to issue the bus use
20 right. As a consequence, the route node arbitrates the requests for the bus use rights from these nodes. A node which acquires the bus use right as a result of the arbitration designates the transmission rate before beginning data transfer. That is, the node informs the
25 transmission destination node that the transmission rate is 100, 200, or 400 Mbps.

Thereafter, in the case of isochronous

00022979-021298
00022979-021298

communication, the transmission source node starts data transfer by the designated channel immediately after receiving a cycle start packet transmitted by the route node as a cycle master in synchronism with the communication cycle. Note that the cycle master transmits the cycle start packet onto the bus and also matches the time of the individual nodes.

In the case of asynchronous communication in which control data such as a command is transferred, on the other hand, after synchronous transfer in each communication cycle is complete, arbitration for asynchronous communication is performed, and data transfer from the transmission source node and the transmission destination node is started.

In addition to the IEEE1394 standard described above, the RS-232C standard and the RS-422 standard presently exist and are used as the conventional serial data communication methods. These standards assume mutual connection using serial binary data exchange between a data terminal equipment (DTE) and a data circuit-terminating equipment (DCE). These standards are formed and open to the public by the American National Standards Institute (ANSI).

As another example of the digital I/F bus systems, a universal serial bus (to be referred to as a USB hereinafter) as defined in Universal Serial Bus Specification (Revision 1.0, January 15 1996) is

09022979-021298
062720-622060

proposed. This bus is invented as an external bus for connecting a PC and its peripheral devices. An outline of this USB system will be described below.

5 The connection form of the USB will be described with reference to Fig. 3. This USB system comprises a host computer 300 such as a PC, a route hub 302, a first device 304 which is a recording medium such as a hard disk, a composite device 306 such as a camera-integrated VTR, a first hub 308, a second device 10 310 such as a video camera, a third device 312 such as a VTR, a second hub 314, a fourth device 316 which is an input device such as a keyboard, and a fifth device 318 which is a pointing device such as a mouse. A hub has a function of adding a USB device. Also, a device 15 is a terminal equipment including a USB bus interface (not shown). In this USB as shown in Fig. 3, the terminal equipments are connected via the hubs including the route hub 302 on the host computer 300, thereby forming a multiple star connection.

20 Since the host computer 300 has rights to access the first, second, third, and fourth devices 304, 310, 312, and 316, data exchange between these devices is performed via the host computer 300. Therefore, bus arbitration between the devices is not performed.

25 In the USB, data transfer is performed by a frame whose unit is 1 ± 0.05 ms. Fig. 4 shows the structure of the frame in the USB. Packets are packed in this

862120" 62622050

frame in accordance with the purpose and transferred.
Four types of packets are defined in the USB. The
first one is a token packet, the second one is a
start-of-frame packet (to be referred to as an SOF
5 packet hereinafter), the third one is a data packet,
and the fourth one is a handshake packet. The frame is
started by the SOF packet.

The host computer 300 performs data transfer with
a plurality of devices by sequentially sending data
10 transfer requests previously scheduled in the frame.
If data is large-amount data, such as image data, which
cannot be contained in a single frame, the host
computer 300 divides the data in units of frames and
transfers the divided data.

15 Packet fields are packed in the above four types
of packets in accordance with the purpose and
transferred. In the USB, six types of packet fields
are defined. The first one is an 8-bit packet
identifier field (to be referred to as a PID
20 hereinafter), the second one is a 7-bit address field
(to be referred to as an ADDR hereinafter), the third
one is a 4-bit endpoint field (to be referred to as an
ENDP hereinafter), the fourth one is an 11-bit frame
number field, the fifth one is a 1- to 1023-byte data
25 field, and the sixth one is a 5- or 16-bit cyclic
redundancy checks field (the 5- and 16-bit ones will be
referred to as a CRC5 and a CRC16, respectively,

09022979-021298

hereinafter). The four types of packets described above are constituted by combining these packet fields.

Figs. 5A to 5D show the arrangements of the four types of packets. As shown in Fig. 5A, the token packet is constituted by the combination of the PID, ADDR, ENDP, and CRC5 fields. As shown in Fig. 5B, the SOF packet is constituted by the combination of the PID, frame number, and CRC5 fields. As shown in Fig. 5C, the data packet is constituted by the combination of the PID, data, and CRC16 fields. As described above, the data field has 1- to 1023-byte data. Also, as shown in Fig. 5D, the handshake packet is constituted only by the PID.

In the USB, two transfer modes are defined. One is a full-speed transfer mode whose average bit rate is 12 Mbps. The other is a low-speed transfer mode whose average bit rate is 1.5 Mbps.

Also, four data transfer methods are defined in the USB. The first one is isochronous transfer. In isochronous transfer, a transfer width which is a data amount of transfer performed for each frame and a transfer time from transfer request to transfer start are guaranteed. Also, in isochronous transfer, no retransmission request can be made even if an error occurs in transfer data. The second one is interrupt transfer. In interrupt transfer, only inputs from the individual devices to the host computer 300 are

09022979.021298

possible. Also, in interrupt transfer, the data transfer priority order on the bus is comparatively high. The third one is bulk transfer. In bulk transfer, the data transfer priority order is the lowest of the four transfer methods. The fourth one is control transfer. Control transfer is performed to exchange setup data for setting up the individual devices.

The 1394 serial bus system and the USB described above are communication systems which have not been put into practical use until recently, and the conventional communication systems using RS-232C and RS-422 are still extensively used presently.

The present situation, therefore, is that all of digital devices corresponding to the 1394, digital devices corresponding to the USB, and digital devices corresponding to RS-232C and RS-422 coexist.

Accordingly, it is expected that apparatuses including the interfaces of both the 1394 and the USB which are main streams in recent years will be extensively demanded. It is also expected that apparatuses including both the 1394 interface and the interface of RS-232C or RS-422 will be extensively demanded. Furthermore, it is expected that apparatuses including a plurality of interfaces of, e.g., the 1394, the USB, and RS-232C will be extensively demanded.

If, however, two or more types of communication

09022979.021298
862120"622060

devices are incorporated into a single apparatus to perform communication by two or more communication systems, the circuit scale is increased, and this significantly increases the cost.

5 SUMMARY OF THE INVENTION

The present invention has been made in consideration of the above situation, and has as its first object to provide a communication apparatus and a communication method capable of selecting two or more
10 communication systems by using a single device without increasing the cost due to an increase in the circuit scale or deteriorating the operability in setting device connection.

To achieve the above object, according to one
15 preferred embodiment of the present invention, there are provided a communication apparatus and a communication method of transmitting and receiving command data for controlling a device connected to a communication line by selecting an arbitrary one of a
20 plurality of different communication systems, wherein at least some of a plurality of command data of each of the different communication systems are used in all of the communication systems.

According to another preferred embodiment of the
25 present invention, there are provided a communication apparatus and a communication method of transmitting and receiving command data for controlling a device

09022979-021298

connected to a communication line by selecting an arbitrary one of a plurality of different communication systems and, on the basis of the received command data, generating control data for a device connected to the communication line, wherein at least some of a plurality of command data generated by each of the different communication systems are used in all of the communication systems.

According to still another preferred embodiment of the present invention, there are provided a communication apparatus and a communication method comprising a first communication mode which performs data communication on a first communication line on the basis of a first communication system, and a second mode which performs data communication on a second communication line on the basis of a second communication system different from the first communication system, wherein the communication modes are switched in accordance with a connection state with respect to the first communication line for the first communication system.

Other objects, features and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a view showing the connection form of an

09022979.021298
862120.62622060

Fig. 2 is a timing chart showing a communication example using the IEEE1394 serial bus;

Fig. 4 is a view showing a data transfer unit of the USB;

10 Fig. 6 is a block diagram showing the arrangement
of a digital video camera of the first embodiment
according to the present invention;

15 Fig. 8 is a flow chart showing the flow of digital
I/F switching control according to the present
invention;

Figs. 10A and 10B are views showing data formats transmitted from a host in bulk transfer;

25 Fig. 12 is a view showing the arrangement of a
data packet in this embodiment; and

Fig. 13 is a block diagram showing the arrangement

of a digital video camera using the data packet shown in Fig. 12.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The first embodiment of the present invention will be described below with reference to the accompanying drawings. Fig. 6 is a block diagram showing the embodiment in which the present invention is applied to a video camera, i.e., a so-called SD (Standard Definition) video camera, for recording and reproducing an SD video signal.

Referring to Fig. 6, this SD video camera comprises a lens 1, an image sensing device 3 such as a CCD, a camera processing unit 5, a recording medium 7 such as a magnetic tape, a helical scan head (to be referred to as a head hereinafter) 9, an error correction circuit (to be referred to as an ECC hereinafter) 11, a video signal processing circuit 13, a switch circuit 15, an audio signal processing circuit 17, and a subcode data processing circuit 19.

The SD video camera further comprises an auxiliary data processing circuit (to be referred to as an AUX data processing circuit hereinafter) 21, an arithmetic processing unit (to be referred to as an MPU hereinafter) 23, a format circuit 25, an interface circuit (to be referred to as an I/F circuit hereinafter) 27, a read-only memory (to be referred to as a ROM hereinafter) 29, a 1394 driver 31, an RS-232C

862120" 62622060

driver 33, a 1394 I/O port 35, an RS-232C I/O port 37, a servo circuit 39, a data bus 41, and a mode controller 43.

5 An object image taken through the lens 1 is photoelectrically converted by the CCD 3 and subjected to predetermined signal processing by the camera processing unit 5. Consequently, a luminance signal Y and color difference signals V and U are generated at a ratio of 4 : 1 : 1 as digital video signals. These
10 digital video signals thus generated are input to the switch circuit 15.

In performing encoding, the digital video signals are applied from the switch circuit 15 to the video signal processing circuit 13 under the switching
15 control by the MPU 23. The video signal processing circuit 13 performs compression coding for the 4 : 1 : 1 digital video signals by block formation, discrete cosine transform (to be referred to as DCT hereinafter), quantization, and fixed-length coding.

20 Also, in performing encoding, a digital audio signal is input from a circuit (not shown) such as a microphone or an audio amplifier to the audio signal processing circuit 17 via the switch circuit 15 and encoded by the audio signal processing circuit 17.
25 Additionally, subcode data and AUX data are input from the MPU 23 to the subcode data processing circuit 19 and the AUX data processing circuit 21, respectively,

09022979 021298
862120 62622060

and processed by these circuits.

The video signal, audio signal, subcode data, and AUX data processed by the video signal processing circuit 13, the audio signal processing circuit 17, the subcode data processing circuit 19, and the AUX data processing circuit 21, respectively, are input to the ECC 11 through the data bus 41. The ECC 11 adds an error correcting code to these signals. The signals are then transmitted through a modulation circuit and a head amplifier (neither is shown) and written on the magnetic tape 7 with the head 9.

In performing decoding, on the other hand, the head 9 reproduces digital signals from a track on the magnetic tape 7. The ECC 11 performs error correction for the reproduced digital signals. Of the digital signals output as digital block data from the ECC 11 to the data bus 41, the video signal processing circuit 13 connected to the data bus 41 decodes a video signal to generate a luminance signal Y and color difference signals U and V at a ratio of 4 : 1 : 1. These generated signals are output outside via the switch circuit 15.

Of the digital signals output as digital block data from the ECC 11, an audio signal is input, similar to the video signal, to the audio signal processing circuit 17 through the data bus 41. The audio signal is decoded by the audio signal processing circuit 17

09022979-021298

and output outside via the switch circuit 15.

Meanwhile, the subcode data processing circuit 19 and the AUX data processing circuit 21 connected to the data bus 41 input decoded subcode data and AUX data, respectively, to the MPU 23.

The compression-coded video and audio data are input to the format circuit 25 through the data bus 41. While the video signal processing circuit 13 and the audio signal processing circuit 17 are performing encoding, the input video and audio data to the format circuit 25 are data before the error correcting code is added by the ECC 11. On the other hand, while the video signal processing circuit 13 and the audio signal processing circuit 17 are performing decoding, the input video and audio data to the format circuit 25 are data after the error correcting code is removed by the ECC 11.

The output subcode data and AUX data from the MPU 23 are also input to the format circuit 25. The format circuit 25 reconstructs these video data, audio data, subcode data, and AUX data into DIF data (digital interface data) and outputs these DIF data to the I/F circuit 27. These DIF data are packeted by the I/F circuit 27.

Note that the format circuit 25 and the I/F circuit 27 are so controlled by the MPU 23 as to perform processing suited to the selected one of the

09022979.021298

1394 interface and the RS-232C interface.

When the 1394 interface is to be used, the packet data formed by the I/F circuit 27 is supplied to the 1394 I/O port 35 via the 1394 driver 31. When the
5 RS-232C interface is to be used, the packet data formed by the I/F circuit 27 is supplied to the RS-232C I/O port 37 via the RS-232C driver 33.

The 1394 driver 31 monitors the state of connection to the 1394 serial bus by detecting the
10 power supply voltage of a power supply twisted pair line of 1394 twisted pair lines, and outputs data indicating the connection state (to be referred to as 1394 connection state data hereinafter) to the MPU 23. The power supply voltage level of the power supply
15 twisted pair line is raised when the 1394 serial bus is connected to the 1394 driver 31, and is lowered when the bus is disconnected. Therefore, the connection state can be detected by monitoring this voltage level.

The 1394 interface and the RS-232C interface can
20 also be switched by the user by operating an external select switch (not shown) or automatically switched by, e.g., the MPU 23 by detecting the connections of the interfaces.

Although the automatic switching between the
25 interfaces will be described later, the mode controller 43 switches the communication modes (switches the communication systems IEEE1394 and RS-232C) in

09022979.021298
862120 62622060

accordance with a setting signal (to be described later) sent from the MPU 23.

The servo circuit 39 controls the running of the magnetic tape 7 in accordance with a designation signal from the MPU 23. Note that the MPU 23 performs processing in accordance with input designating information from an operation panel (not shown) and manages the operation mode of the whole system of this digital VTR and its various status transitions.

This servo circuit 39 primarily has a function of stationarily maintaining the driving of a rotary drum and a capstan (neither is shown). That is, the servo circuit 39 is connected to a capstan motor (not shown) for controlling the tape feed speed, a capstan FG (Frequency Generator) for checking the rotating state of the capstan motor, a drum motor for rotating a rotary drum, and detectors FG and PG (Phase Generator) for checking the rotational speed and the rotational phase of the drum motor. These components are controlled by the servo circuit 39.

Command data for the SD video camera of this embodiment is externally applied to the 1394 I/O port 35. Figs. 7A and 7B are views showing general formats of the 1394 command data applied to the 1394 I/O port 35. Referring to Figs. 7A and 7B, CT and RC are 4-bit codes indicating a command type and a response code, respectively. Table 1 below shows codes of the command

09022979 021298
B62120 62622060

type, and Table 2 below shows codes of the response code.

[Table 1]

CT/RC code (binary)				Command type
MSB		LSB		
0	0	0	0	Control command
0	0	0	1	State inquiry command
0	0	1	0	Support inquiry command
0	0	1	1	Report request command
0	1	0	0	(Unused)
0	1	0	1	(Unused)
0	1	1	0	(Unused)
0	1	1	1	(Unused)

09022979.021298
1862120.1622060

[Table 2]

CT/RC code (binary) MSB LSB	Response code
1 0 0 0	Conditions unfulfilled
1 0 0 1	Admitted
1 0 1 0	Rejected
1 0 1 1	Transiting
1 1 0 0	Conditions fulfilled/standby
1 1 0 1	Already changed
1 1 1 0	(Unused)
1 1 1 1	Busy

Referring to Figs. 7A and 7B, HA indicates a header address, and EHA indicates an extended header address. The header address is an 8-bit code and used as an identification code for a plurality of subdevices in one device connected to a communication interface (communication line). That is, the five upper bits of the header address indicate a subdevice type representing the type of the subdevice, and the three lower bits of the header address indicate a subdevice number representing the number of the subdevice among subdevices of the same type indicated by the five upper

09022979-021298
862120-622060

bits. The extended header address is a header address reserved for the future. Table 3 below shows examples of the subdevice type.

[Table 3]

Code (binary)					Subdevice type
MSB				LSB	
0	0	0	0	0	Video monitor
0	0	0	0	1	(Unused)
0	0	0	1	0	(Unused)
0	0	0	1	1	(Unused)
0	0	1	0	0	Video cassette recorder (VCR)
0	0	1	0	1	TV tuner
0	0	1	1	0	(Unused)
0	0	1	1	1	Video camera
0	1	0	0	0	(Unused)
		.			
		.			
		.			
1	1	1	1	1	

Referring to Figs. 7A and 7B, OPC indicates an operation code, and OPR indicates an operand. The operation code indicates the contents of control with respect to a digital device connected to a

00022979.021298
862T20" 6/622060

communication interface (communication line). The operand indicates data required by the operation code. Table 4 below shows examples of the operation code and operand for reproduction.

[Table 4]

OPC		OPR	
Reproduction	0xC3	Next frame	0x30
		Lowest rate	0x31
		Low rate 4	0x32
		Low rate 3	0x33
		Low rate 2	0x34
		Low rate 1	0x35
		Normal rate (x1)	0x36
		High rate 1	0x37
		High rate 2	0x35
		High rate 3	0x38
		High rate 4	0x39
		Highest rate	0x3A
		Preceding frame	0x3B

The unit of the data length of a command shown in Figs. 7A and 7B is four bytes. If the data length does not reach an integer multiple of four bytes, data in

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

which all bits are zero is packed in the end of a bit stream so that the data length is an integral multiple of four bytes as a whole.

When the user intends to cause the video cassette recorder (VCR) to perform normal reproduction in the case where the 1394 command data shown in Figs. 7A and 7B are represented as shown in Tables 1 to 4, a code of command data such as "0x0021C336" (0x indicates hexadecimal notation) is input from an external device. That is, the first "0x00" indicates four bits fixed to 0 at the beginning of the command data and subsequent four bits (see Table 1) representing that the command data is a control command. The next "0x21" indicates an 8-bit header address and shows that the subdevice type is a VCR (see Table 3) and the subdevice is the second device in the VCR. The next "0xC336" indicates an operation code and an operand obtained from Table 4.

When this code for normal reproduction is input to the 1394 I/O port 35, on the basis of this input code the I/F circuit 27 generates an address in the ROM 29 storing control data for normal reproduction and applies the address to the MPU 23. In accordance with the generated address, the MPU 23 reads out the control data from the ROM 29 and controls the rotary drum and the capstan motor (neither is shown) via the servo circuit 39, thereby holding the reproduction state.

Meanwhile, command data for RS-232C is externally

09022979-021298
862120" 622060

input to the RS-232C I/O port 37. In this embodiment, at least some command data of the 1394 command data and the RS-232C command data are used in the both systems.

For example, command data received by the 1394 driver 31 and the RS-232C driver 33 by their respective communication systems and having the same function are used in the two systems. More specifically, assume that M (= integer of 2 or more) command data are transmitted and received by RS-232C and N (= integer of 2 or more, $N \geq M$) command data including command data having the same functions as the M command data are transmitted and received by the 1394. If this is the case, all of the M command data are used in both the 1394 and RS-232C. Alternatively, some of the M command data are used in the two systems.

With this arrangement, a communication device including the I/F circuit 27, the MPU 23, and the like which perform various control operations by interpreting command data can be shared by the 1394 communication system and the RS-232C communication system. This eliminates the need to provide a plurality of communication devices for various communication systems in one digital device. Accordingly, a digital device corresponding to both the 1394 and RS-232C communication systems can be manufactured without increasing the circuit scale.

In RS-232C, two digital devices are usually

09022979 021298
862720 6222060

connected in a one-to-one correspondence with each other. This makes device identification codes, device numbers, and the like data unnecessary. In this embodiment, therefore, a code such as "0xC336" is input
5 to the RS-232C I/O port 37 to indicate the same normal reproduction. Table 5 below shows examples of the RS-232C reproduction codes corresponding to the 1394 control codes described above.

09022979.021298

[Table 5]

Reproduction code	
Next frame	0xC330
Lowest rate	0xC331
Low rate 4	0xC332
Low rate 3	0xC333
Low rate 2	0xC334
Low rate 1	0xC335
Normal rate (×1)	0xC336
High rate 1	0xC337
High rate 2	0xC335
High rate 3	0xC338
High rate 4	0xC339
Highest rate	0xC33A
Preceding frame	0xC33B

When data is transferred by omitting the identification code and the device number of a device as described above, a delay time caused by command transfer can be reduced. This is convenient when RS-232C which is a relatively-low-rate interface is used.

When this code for normal reproduction is input to

the RS-232C I/O port 37, on the basis of this input code the I/F circuit 27 generates an address in the ROM 29 storing control data for normal reproduction and inputs the address to the MPU 23. In accordance with the generated address, the MPU 23 reads out the control data from the ROM 29 and controls the rotary drum and the capstan motor (neither is shown) via the servo circuit 39, thereby holding the reproduction state.

In the above embodiment, the command data applied to the 1394 I/O port 35 and the RS-232C I/O port 37 and having the same function are the same in the two communication systems. However, even when these command data are different, an increase in the circuit scale can be prevented by generating common control data in the two communication systems from the ROM 29 on the basis of the command data.

If this is the case, the I/F circuit 27 and the MPU 23 generate the same control data for command data received by the 1394 I/O port 35 and the RS-232C I/O port 37 by their respective communication systems and having the same function. That is, the I/F circuit 27 which generates addresses in the ROM 29 storing control data corresponding to the command data received by the I/O ports 35 and 37 generates the same address in the ROM 29 for the command data received by the two communication systems and having the same function.

The above embodiment is described by using the

09022979-021298
862120-6222060

IEEE1394 standard and the RS-232C standard. However, some other standard (e.g., the RS-422 standard) can also be used. Also, when common command data is used for not only a communication apparatus corresponding to two communication standards but also a communication apparatus corresponding to a larger number of communication standards, the communication apparatus can be manufactured without increasing the circuit scale.

10 In the above embodiment, the two lower bytes of the control command are the same in the IEEE1394 standard and the RS-232C standard. However, this common part can have another arrangement. Additionally, the code length for control is not limited to the above-mentioned code length (four bytes), so any arbitrary code length can be applied.

15 In this embodiment as described above, in an apparatus in which a given one of a plurality of different communication systems is selected to transmit and receive command data for controlling devices connected to a communication line, at least some of a plurality of command data of the different communication systems or of a plurality of device control data generated on the basis of the received command data are used in all of these communication systems. Therefore, a common communication apparatus for performing various control operations by

09022979.021298
852120.622060

5 communication systems in one device. Consequently, it is possible to provide a communication apparatus which can select two or more different communication systems and does not largely increase the cost due to an increase in the circuit scale.

As described above, the 1394 driver 31 inputs the 1394 connection state data to the MPU 23 in addition to the control data from the 1394 serial bus. If data indicating that the 1394 serial bus is connected is input, the MPU 23 supplies a 1394 setting signal to the mode controller 43 in order to set the communication mode in the 1394 mode. Upon receiving the 1394 setting signal, the mode controller 43 holds the 1394 driver 31 active and holds the RS-232C driver 33 in sleep.

On the other hand, if the 1394 driver 31 is disconnected from the 1394 serial bus, the power supply voltage of the power supply twisted pair line of the 1394 twisted pair lines drops. The 1394 driver 31
25 detects this voltage drop and outputs to the MPU 23 data indicating that the 1394 driver 31 is disconnected from the 1394 serial bus. Upon receiving this 1394

connection state data, the MPU 23 supplies an RS-232C setting signal to the mode controller 43 in order to set the communication mode in the RS-232C mode. When this RS-232C setting signal is input, the mode
5 controller 43 sets the 1394 driver 31 in sleep and the RS-232C driver 33 active.

Fig. 8 is a flow chart showing the flow of interface switching control. Referring to Fig. 8, the control is started from step 0. In step 1, the
10 communication mode is reset to the 1394 mode. In step 2, the 1394 driver 31 is set active. In step 3, the RS-232C driver 33 is set in sleep.

In step 4, the level of the power supply voltage of the power supply twisted pair line of the 1394
15 twisted pair lines is compared with a threshold voltage Th previously determined in the system to check whether the power supply voltage is higher than the threshold voltage Th . This threshold voltage Th is set to, e.g., 4 V.

20 If the power supply voltage of the power supply twisted pair line of the 1394 twisted pair lines is higher than the threshold voltage Th , the flow returns to step 2. If this is the case, the processes in steps 2 to 4 form a loop to hold the 1394 driver 31 active
25 and the RS-232C driver 33 in sleep.

On the other hand, if it is determined in step 4 that the power supply voltage of the power supply

09022979-021298
B62120-6222060

twisted pair line of the 1394 twisted pair lines is lower than the threshold voltage T_h , the flow advances to step 5. In step 5, the 1394 driver 31 is set in sleep. In step 6, the RS-232C driver 33 is set active. In step 7, the communication mode is set in the RS-232C mode. In final step 8, the control is completed.

Although not shown in the flow chart of Fig. 8, in this embodiment the power supply voltage of the power supply twisted pair line of the 1394 twisted pair lines is measured at fixed time intervals. If this measured power supply voltage is higher than the threshold voltage T_h previously set in the system, the control start routine in step 0 is started. Therefore, even after the communication mode is set in the RS-232C mode, if the 1394 driver 31 is again connected to the 1394 serial bus, the communication mode is automatically switched to the 1394 mode.

In this embodiment as described above, it is unnecessary to provide a plurality of communication apparatuses for various communication systems in one digital device. Therefore, a digital device corresponding to two communication systems of the 1394 and RS-232C can be manufactured without increasing the circuit scale.

In this embodiment, data communication is performed in the 1394 communication mode when the 1394 driver 31 is connected to the 1394 serial bus. When

09022979-0212998

the 1394 serial bus is disconnected, the RS-232C driver 33 is automatically activated to perform data communication in the RS-232C communication mode.

Therefore, even when a plurality of devices are

5 connected to both the 1394 serial bus and the RS-232C data channel, no connection setting operation need be performed, and high operability can be realized.

Although the above embodiment is described by

10 using the IEEE1394 standard and the RS-232C standard, another standard (e.g., the RS-422 standard) can also be used in place of the RS-232C standard.

Alternatively, a communication line of any other standard than the IEEE1394 standard can be used, provided that the channel has a function of supplying
15 power.

The second embodiment according to the present invention will be described below with reference to the accompanying drawings. Fig. 9 is a block diagram showing the embodiment in which the present invention
20 is applied to a video camera, i.e., a so-called SD (Standard Definition) camera, for recording and reproducing an SD video signal. The second embodiment differs from the first embodiment in that the first embodiment includes the IEEE1394 and RS-232C as digital
25 interfaces, but the second embodiment includes the IEEE1394 and the USB. That is, this embodiment discloses an apparatus in which at least some of

09022979.021298

command data or of a plurality of device control data generated on the basis of the received command data are used in communication systems of both the 1394 and the USB. The same reference numerals as in Fig. 6 denote the same parts in Fig. 9, and a detailed description thereof will be omitted.

Referring to Fig. 9, this apparatus comprises a bit inversion circuit 45, a USB driver 47, and a USB I/O port 49.

Video and audio DIF data are directly input from a data bus 41 to a format circuit 25'. The format circuit 25' also receives subcode data and AUX data from an MPU 23, converts the data into DIF data, and outputs the DIF data. This DIF data is packeted by an I/F circuit 27'.

The format circuit 25' and the I/F circuit 27' are so controlled as to perform processing suited to one of the 1394 interface and the USB interface selected by the MPU 23.

When the 1394 interface is to be used, data is supplied to a 1394 I/O port 35 via a 1394 driver 31. When the USB interface is to be used, data is supplied to the bit inversion circuit 45, the USB driver 47, and the USB I/O port 49.

The 1394 and the USB transmit data by using different bit output methods; i.e., the most significant bit is transmitted first (MSB first) in the

09022979-021298
862720-6222060

1394, and the least significant bit is output first (LSB first) in the USB. When the USB is used, therefore, the bit inversion circuit 45 performs bit inversion for data to be transmitted and received. In this embodiment, bit inversion is performed for data to be transmitted and received by the USB on the basis of the 1394. However, bit inversion can also be performed for data to be transmitted and received by the 1394 on the basis of the USB.

The 1394 interface and the USB interface can be switched by the user by using an external select switch or can also be automatically switched by detecting the connections of these interfaces.

The automatic interface switching will be described later.

As shown in the first embodiment, a code such as "0x0021C336" is input from an external device when the user intends to cause a VCR to perform normal reproduction.

When this normal reproduction code is input from the 1394 I/O port 35, on the basis of the input code the I/F circuit 27' generates an address in a ROM 29 storing data for normal reproduction and applies the address to the MPU 23. In accordance with this address data, the MPU 23 reads out the control data from the ROM 29 and controls a rotary drum and a capstan motor (neither is shown) via a servo circuit 39, thereby

09022979-021298
862120-622060

holding the reproduction state.

On the other hand, when the user intends to perform normal reproduction in the same manner as above by using the USB, a code is transmitted from a host (not shown) by the bulk transfer described earlier. In the bulk transfer, a token packet as shown in Fig. 10A is transmitted in the start frame of the bulk transfer. In the next frame, the host transmits a data packet as shown in Fig. 10B. A code such as 0x0021C336 described above is contained in a data field of this data packet. Transmission of such comparatively small data is complete by two frames because data up to 1,023 bytes can be contained in the data field.

[Table 6] PIDs of token packet

PID name	PID value
OUT	11100001 ₂
IN	01101001 ₂
SETUP	00101101 ₂

[Table 7] PIDs of data packet

PID name	PID value
DATA0	11000011 ₂
DATA1	01001011 ₂

The PIDs in the token packet will be described

5

10

20

25

When the transmission from the host is complete, the SD video camera of this embodiment informs the host of the completion of the transmission by using a handshake packet as shown in Fig. 11. This handshake packet is constituted only by PID and changes the meaning of information in accordance with the value of the PID. Table 8 below shows the values of the PID. ACK indicates that the communication is normally complete. NACK indicates that the data from the host has an error. If this is the case, the host repeats the same data transfer as above. STALL indicates that the SD video camera of this embodiment is made unable to perform data transmission/reception for some reason.

[Table 8] PIDs of handshake packet

PID name	PID value
ACK	11010010 ₂
NACK	01011010 ₂
STALL	00011110 ₂

When informed by ACK that the communication is normally completed, the host again transmits the token packet to the SD video camera of this embodiment in order to receive a response code. In the next frame, the SD video camera of this embodiment inserts, e.g., a response code 0x0921C336, indicating that normal reproduction is possible, into a data packet and

transmits the data packet. When normally receiving the data packet, the host transmits the ACK handshake packet to the SD video camera of this embodiment to complete one transmission/reception.

5 In this embodiment, the control code and the response code are exchanged by using only bulk transfer. However, USB interrupt transfer can also be used in transmitting the response code. This configuration has the advantage that the response code
10 can be reliably returned even if the data amount on the communication line is increased, since interrupt transfer has a higher transfer priority order than that of bulk transfer. Also, isochronous transfer can be used instead of bulk transfer in exchanging the control
15 code and the response code.

 USB communication can also be performed by adding additional information to a data packet for performing the communication. Fig. 12 shows a whole data packet when the additional information is added, and the
20 arrangement of a data field in the data packet. This data packet is transmitted by, e.g., isochronous transfer. During the transmission, the data field of the data packet is transmitted by a fixed length from the start to the end of the communication.

25 In the data field shown in Fig. 12, a field to be transmitted first is a data_length field. The data length of the data_length field is set to, e.g., 1

00022979-021298
862720" 6222000

byte. This field indicates an effective data length
contained in the data field in units of bytes. A
fixed_length_data_field follows the data_length field.
This field is a fixed-length data field as described
5 above. This field includes a valid_data field
containing effective data to be actually decoded and a
zero_pad_byte field in which data whose value is 0 is
packed. If the data length of the valid_data field
equals the data length of the fixed_length_data_field,
10 the zero_pad_byte field does not exist in the
fixed_length_data_field. The value of the
zero_pad_byte field is not limited to zero, and some
other data such as 0xFF can also be used. The data
length of the fixed_length_data_field is set to, e.g.,
15 15 bytes.

In the data field shown in Fig. 12, an RS_code
field to be transmitted last is an error
detecting·correcting code such as a Reed-Solomon code.
Although a Reed-Solomon code is used in this
20 embodiment, another error detecting·correcting code
such as a Humming code can also be used. In this
embodiment, a Reed-Solomon code having, e.g., 8 bytes
is added.

Fig. 13 is a block diagram in which a
25 communication apparatus using the data packet shown in
Fig. 12 is applied to an SD video camera. The same
reference numerals as in Fig. 9 denote the same parts

09022979.021288
862120.6222060

in Fig. 13, and a detailed description thereof will be omitted. Referring to Fig. 13, this SD video camera comprises a second error correcting circuit (ECC) 51 and a second data bus 53.

5 When normal reproduction is to be performed, a code in which the value of the valid_data field in the data field shown in Fig. 12 is 0x0021C336 is applied from an external device to the USB driver 47 via the USB I/O port 49. Since the effective data length is
10 set to four bytes, the value of the data_length field is 0x04. In fact, the value of the above data is bit-inverted in units of bytes. The bit inversion circuit 45 converts this bit-inverted value into a normal value. Each converted data is supplied to the
15 second error correcting circuit 51 through the second data bus 53, and errors occurring on the communication line are detected and corrected. In this embodiment, 4-byte correction is possible.

 With the above arrangement, even in isochronous
20 transfer in which a retransmission request cannot be made by NACK, the accuracy of data in communication can be increased. Also, the use of isochronous transfer having a comparatively high priority order has the advantage that the response can be rapidly performed in
25 exchanging data. Note that the data format described above can contain not only the data length and the error detecting·correcting code but also some other

09022979-021298

additional information. Note also that the numbers of bytes of the data_length field, the fixed_length_data_field, and the RS_code field are not limited to those of the above arrangement, so another arrangement can naturally be used. Furthermore, although isochronous transfer is used in this embodiment, the above arrangement is applicable to another transfer system such as bulk transfer.

The automatic interface switching will be described next.

As described in the first embodiment, the 1394 driver 31 monitors the connection of the 1394 by detecting the power supply voltage of a power supply twisted pair line of 1394 twisted pair lines, and outputs the connection state to the MPU 23. The USB driver 47 also monitors the connection of the USB from the signal statuses of USB twisted pair lines and outputs the connection state to the MPU 23.

First, connection switching when the 1394 driver 31 is the master will be described. In this case, 1394 connection is performed as much as possible. The 1394 driver 31 applies a control signal from the 1394 bus and the 1394 connection state data described earlier to the MPU 23. While the 1394 driver 31 is applying to the MPU 23 the data indicating that the 1394 bus is connected, the MPU 23 holds the 1394 driver 31 active in order to set the communication mode in the 1394

05022975-021298
952720-5222050

When the 1394 driver 31 is disconnected from the 1394 bus, the power supply voltage of the power supply twisted pair line of the 1394 twisted pair lines drops. The 1394 driver 31 detects this voltage drop and outputs to the MPU 23 data indicating that the 1394 driver 31 is disconnected from the 1394 bus. When receiving this data, the MPU 23 sets the 1394 driver 31 in sleep and the USB driver 47 active.

Next, connection switching when the USB driver 47 is the master will be described. In this case, USB connection is performed as much as possible. The USB driver 47 applies a control signal from the USB bus and the USB connection state data described previously to the MPU 23. The USB bus signal operates as differential signals. The differential signals are D+ and D-. When the USB driver 47 is connected to the USB bus, one of the D+ and D- holds a voltage higher than a maximum value $V_{SE}(MAX)$ of a single-end threshold, and the other has a voltage lower than $V_{SE}(MAX)$. The USB driver 47 detects this state and applies data indicating USB bus connection to the MPU 23. While the USB driver 47 is inputting, to the MPU 23, the data indicating that the USB bus is connected, the MPU 23 holds the USB driver 47 active in order to set the communication mode in the USB mode. Also, upon

receiving this USB setting signal, the MPU 23 holds the 1394 driver 31 in sleep.

When the USB driver 47 is disconnected from the USB bus, the voltages of both D+ and D- become lower than the voltage $V_{SE}(MAX)$. If this state continues for 2.5 μ sec or longer, the USB driver 47 determines that the connection is cut. The USB driver 47 outputs, to the MPU 23, data indicating that the USB driver 47 is disconnected from the USB bus. Upon receiving this data, the MPU 23 sets the communication mode in the 1394 mode. Also, the MPU 23 sets the USB driver 47 in sleep and the 1394 driver 31 active.

The switching performed to set the 1394 or the USB as the master can also be performed by the user by using a switch (not shown) or the like. Also, the 1394 can be set as the master by a standard operation, and this can be reset when the power supply is turned on. Alternatively, the USB can be set as the master by a standard operation, and this can be reset when the power supply is turned on. Any arbitrary setting can be performed as long as the apparatus operates as above.

The above embodiments have been described by using two digital I/Fs. However, the present invention is similarly applicable to an apparatus including three or more digital I/Fs such as the IEEE1394, the USB, and RS-232C.

362720" 64622060

In other words, the foregoing description of embodiments has been given for illustrative purposes only and not to be constructed as imposing any limitation in every respect.

5 The scope of the invention is, therefore, to be determined solely by the following claims and not limited by the text of the specifications and alterations made within the scope equivalent to the scope of the claims fall within the true spirit and
10 scope of the invention.

09022979.021298
862120 62622060